OPTIMIZATION WITH TENSORS

General problem: Given a loss function $\mathcal{L} : \mathbb{R}^{d_1 \times \cdots \times d_N} \to \mathbb{R}$, we want to solve

$$\min_{W \in \mathbb{R}^{d_1 \times \cdots \times d_N}} \mathcal{L}(W) \quad \text{subject to} \quad \text{rank}(W) \leq R$$

↳ CP rank, Tucker rank, TT rank, ...

Examples of loss functions:

+ Low rank approximation

$$\min_{W} \| W - T \|_F^2 \quad \text{s.t.} \quad \text{rank}_{CP}(W) \leq R$$

↳ the loss function is $\mathcal{L}(W) = \| W - T \|_F^2$

+ Regression: We want to learn a linear function $f : \mathbb{R}^{d_1 \times \cdots \times d_N} \to \mathbb{R}$ from a dataset $\{ (X_1, y_1), (X_2, y_2), \ldots, (X_m, y_m) \} \subseteq \mathbb{R}^{d_1 \times \cdots \times d_N} \times \mathbb{R}$.

f is linear $\Leftrightarrow f(X) = f_W(X) = \langle W, X \rangle = \sum_{i_1, \ldots, i_N} W_{i_1, \ldots, i_N} X_{i_1, \ldots, i_N}$.

In this case, a natural loss function is

$$\mathcal{L}(W) = \sum_{i=1}^{m} \left(f_W(X_i) - y_i \right)^2$$

$$= \sum_{i=1}^{m} \left(\langle W, X_i \rangle - y_i \right)^2$$

+ Completion: Target tensor $T \in \mathbb{R}^{d_1 \times \cdots \times d_N}$

Data: observed entries $\{ T_{i_1, \ldots, i_N} \mid (i_1, \ldots, i_N) \in \Omega \}$

$$\Omega \subseteq [d_1] \times [d_2] \times \cdots \times [d_N]$$

- $\begin{pmatrix} 1 & ? & ? \\ ? & 2 & ? \\ ? & 3 & ? \end{pmatrix}$ $\quad \Omega = \{ (1,1), (2,2), (2,3) \}$

- Image completion

For completion, the loss function is:

$$\mathcal{L}(W) = \sum_{(i_1,\dots,i_N)\in\Omega} \left(T_{i_1\dots i_N} - W_{i_1\dots i_N}\right)^2$$

↳ set of observed entries

Remark: If $\Omega = [d_1]\times[d_2]\times\dots\times[d_N]$, then

$$\sum_{(i_1,\dots,i_N)\in\Omega} \left(T_{i_1\dots i_N} - W_{i_1\dots i_N}\right)^2 = \|T - W\|_F^2$$

- COLLABORATIVE FILTER / RECOMMANDATION SYSTEMS

(NETFLIX CHALLENGE)

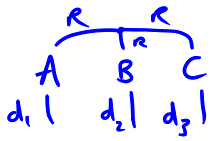
$$\begin{array}{cc} & j \quad \text{movies} \\ \text{users} \; i & \begin{pmatrix} & & & ? & \\ 4 & & ? & & ? \\ & ? & 3 & & ? \\ 2 & & & & \end{pmatrix} \end{array}$$

user i gave rating 4 on movie j

(I) GENERAL OPTIMIZATION ALGORITHM

PBM 1 $\min_{W \in \mathbb{R}^{d_1 \times \dots \times d_N}} \mathcal{L}(W)$ subject to $\mathrm{rank}_{CP}(W) \leq R$

Notation: Given $A_i \in \mathbb{R}^{d_i \times R}$ for $i = 1, \dots, N$, we denote the CP decomposition with factors $A_1, \dots, A_N$ by $CP(A_1, \dots, A_N)$

ex: $CP(A, B, C) =$ 

PBM 1 is equivalent to:

PBM2 $\min_{\substack{A_i \in \mathbb{R}^{d_i \times R} \\ i=1,\dots,N}} \mathcal{L}(CP(A_1, \dots, A_N))$

1) Gradient based algorithm

ALGORITHM: Initialize $A_1, A_2, \dots, A_N$

Repeat

  for $i = 1 \dots N$

   $X_i \leftarrow A_i - \gamma \nabla_{A_i} \mathcal{L}(CP(A_1, \dots, A_N))$ learning rate

  for $i = 1 \dots N$

   $A_i \leftarrow X_i$

until convergence

2) Alternating minimization

We assume $\min_{A_i} \mathcal{L}(CP(A_1, \dots, A_N))$ is easy.

ALGORITHM: Initialize $A_1, A_2, \dots, A_N$

Repeat

  for $i = 1, \dots, N$

   $A_i \leftarrow \arg\min_{A_i} \mathcal{L}(CP(A_1, \dots, A_N))$

until convergence

3) Alternating minimization for low CP rank approximation

Given $T \in \mathbb{R}^{d_1 \times \cdots \times d_N}$, target rank R:

$\boxed{\text{CP-PBM}}$
$$\min_{\substack{A_i \in \mathbb{R}^{d_i \times R} \\ i=1,\dots,N}} \| T - CP(A_1, \dots, A_N) \|_F^2$$

- Matricization: $T \in \mathbb{R}^{d_1 \times \cdots \times d_N}$, for any $m \in [N]$, $T_{(m)} \in \mathbb{R}^{d_m \times d_1 \cdots d_{m-1} d_{m+1} \cdots d_N}$

  "mode-m matricization"

- Remark: $\|T\|_F = \|T_{(m)}\|_F$ for all $m \in [N]$

Déf: If $A \in \mathbb{R}^{m \times n}$ and $B \in \mathbb{R}^{p \times q}$, then Kronecker product $A \otimes B \in \mathbb{R}^{mp \times nq}$ is defined by:

$$A \otimes B = \underbrace{\begin{pmatrix} a_{11} B & \cdots & a_{1n} B \\ \vdots & \ddots & \vdots \\ a_{m1} B & \cdots & a_{mn} B \end{pmatrix}}_{nq} \quad mp$$

(a_{ij} is the entry (i,j) of A)

Remark: $\underbrace{\underset{m\times 1}{a} \otimes \underset{n \times 1}{b}}_{mn \times 1} = \mathrm{vec}(b a^T) = \mathrm{vec}(\underbrace{b \circ a}_{n \times m})$

Déf: The Kathri-Rao product of $A = \begin{pmatrix} | & & | \\ a_1 & \cdots & a_R \\ | & & | \end{pmatrix} \in \mathbb{R}^{m \times R}$ and $B = \begin{pmatrix} | & & | \\ b_1 & \cdots & b_R \\ | & & | \end{pmatrix} \in \mathbb{R}^{n \times R}$ is defined by

$$A \odot B = \begin{pmatrix} | & | & & | \\ a_1 \otimes b_1 & a_2 \otimes b_2 & \cdots & a_R \otimes b_R \\ | & | & & | \end{pmatrix} \in \mathbb{R}^{mn \times R}$$

Property: If $W = CP(A_1, A_2, \dots, A_N)$, then

$$\underset{d_m \times d_N \cdots d_{m+1} d_{m-1} \cdots d_1}{W_{(m)}} = \underset{d_m \times R}{A_m} \big(\underbrace{A_N \odot \cdots \odot A_{m+1} \odot A_{m-1} \odot \cdots \odot A_1}_{d_N \cdots d_{m+1} d_{m-1} \cdots d_1 \times R}\big)^T$$

Alternating least squares algorithm (ALS) for CP decomposition

$$\min_{\substack{A,B,C \\ d_1\times R \;\; d_2\times R \;\; d_3\times R}} \| T - CP(A,B,C)\|_F^2$$

We want solve this problem wrt A:

$$\arg\min_A \| T - CP(A, \underbrace{B, C}_{\text{fixed}})\|_F^2 = \arg\min_A \| T_{(1)} - (CP(A,B,C))_{(1)} \|_F^2$$

$$= \arg\min_A \| T_{(1)} - A (C \odot B)^T \|_F^2$$

$$= T_{(1)} \left[(C \odot B)^T \right]^{+}$$

pseudo-inverse

(side note: if $A \in \mathbb{R}^{m\times n}$ with $m \leq n$ and $\operatorname{rank}(A) = m$, then

$AA^+ = I$ (however: ⚠ A^+A may not be I ⚠))

ALS ALGORITHM:

INPUT : target tensor T ($d_1 \times \cdots \times d_N$), rank R

OUTPUT : factor matrices A_i, $i=1,\dots,N$ s.t.

$$T \approx CP(A_1, \dots, A_N)$$

- Initialize $A_1, \dots, A_N$
- Repeat

  For $n = 1, \dots, N$

  $$A_n \leftarrow T_{(n)} \left[\left(A_N \odot \cdots \odot A_{n+1} \odot A_{n-1} \odot \cdots \odot A_1 \right)^T \right]^{+}$$

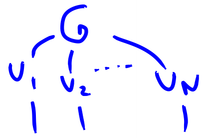
  until convergence

II SVD-based algorithms for Tucker and TT

1) Tucker: Higher order SVD (Lauthawer et al. 2000)

Given $\mathcal{T} \in \mathbb{R}^{d_1 \times \dots \times d_N}$ and $(R_1, \dots, R_N)$ we want to solve

$$\min_{\substack{\mathcal{G} \in \mathbb{R}^{R_1 \times \dots \times R_N} \\ U_i \in \mathbb{R}^{d_i \times R_i}, \\ i=1 \dots N}} \| \mathcal{T} - \underbrace{\mathcal{G} \times_1 U_1 \times_2 U_2 \times \dots \times_N U_N}_{} \|_F^2$$



HOSVD Algorithm

INPUT : $\mathcal{T} \in \mathbb{R}^{d_1 \times \dots \times d_N}$ and $(R_1, \dots, R_N)$

OUTPUT : $\mathcal{G} \in \mathbb{R}^{R_1 \times \dots \times R_N}$, $U_i \in \mathbb{R}^{d_i \times R_i}$, $i = 1 \dots N$ s.t. $\mathcal{T} \approx \mathcal{G} \times_1 U_1 \times_2 \dots \times_N U_N$

For $m = 1 \dots N$

$\quad \underbrace{U_m}_{d_m \times R_m} \leftarrow R_m$ leading left singular vectors of $\mathcal{T}_{(m)}$ $\xrightarrow{\text{SVD}} U D V^T \rightarrow$ extract the first R_m columns

$$\underbrace{\mathcal{G}}_{R_1 \times \dots \times R_N} \leftarrow \underbrace{\mathcal{T}}_{d_1 \times \dots \times d_N} \times_1 \underbrace{U_1^T}_{R_1 \times d_1} \times_2 \dots \times_N \underbrace{U_N^T}_{R_N \times d_N}$$

RETURN $\mathcal{G}, U_1, \dots, U_N$

$$\star \quad \min_{\mathcal{X}} \| \mathcal{T} - \mathcal{X} \|_F^2 \quad \text{s.t.} \quad \operatorname{rank}_{\text{TUCKER}}(\mathcal{X}) \leq (R_1, \dots, R_N)$$

Theorem: Let $\mathcal{X}^*$ be the solution of problem $\star$ and $\mathcal{X}_{HOSVD}$ the solution returned by HOSVD.
$\hookrightarrow \mathcal{G} \times_1 U_1 \times \dots \times U_N$

Then

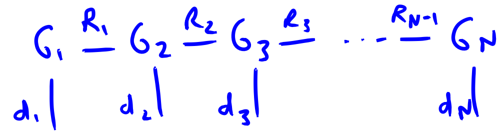
$$\| \mathcal{T} - \mathcal{X}_{HOSVD} \|_F \leq \sqrt{N} \, \| \mathcal{T} - \mathcal{X}^* \|_F$$

- If $\mathcal{T}$ has Tucker rank less than $(R_1, \dots, R_N)$, then $\| \mathcal{T} - \mathcal{X}^* \|_F = 0$ hence $\| \mathcal{T} - \mathcal{X}_{HOSVD} \|_F = 0$ $\rightsquigarrow$ HOSVD recover the exact Tucker decomposition if it exists.

2) Tensor Train : TT-SVD (Ossedelets, 2010)

↳ (a.k.a. Matrix Product State (MPS) : sequential SVD)

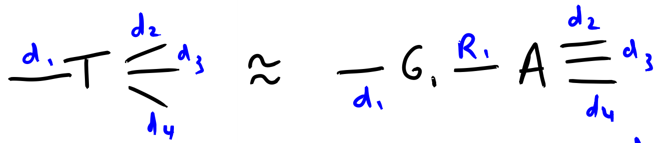
$$\min_{G_1, G_2, \dots, G_N} \| T - TT(G_1, G_2, \dots, G_N) \|_F^2$$

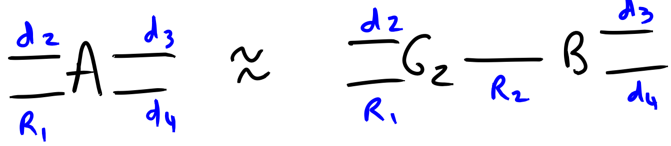


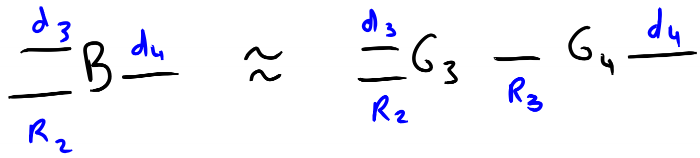
TT-SVD Algorithm

INPUT : $T \in \mathbb{R}^{d_1 \times d_2 \times d_3 \times d_4}$, $(R_1, \dots, R_3)$

OUTPUT : $G_1, \dots, G_4$ s.t. $T \approx TT(G_1, G_2, G_3, G_4)$

- 

(Low rank matrix approx. → SVD)
- 

- 

- RETURN G_1, G_2, G_3, G_4

<u>Theorem</u> : If X^* is the best approximation of T of TT rank $(R_1, \dots, R_{N-1})$ then $\| T - X_{TT-SVD} \|_F \leq \sqrt{N-1} \, \| T - X^* \|_F$.

↳ solution returned by TT-SVD

$$X_{TT-SVD} = G_1 - G_2 - \cdots - G_N$$

- If $\| T - X^* \|_F = 0$ then TT-SVD returns an exact TT decomposition of T.

(III) SUPERVISED LEARNING WITH TENSOR NETWORKS

Supervised Learning With Quantum-Inspired Tensor Networks (NeurIPS, 2016)

E. Miles Stoudenmire[1,2] and David J. Schwab[3]

We want to learn a linear function $f : \mathbb{R}^{d^m} \longrightarrow \mathbb{R}^{h}$

input space is very high dimension

$f : x \longmapsto Wx$

W: $h \times d^m$

Feature map : Image $\longrightarrow \mathbb{R}^{d \times d \times \dots \times d}$

$$x \in \mathbb{R}^{h \times w} \longmapsto \underbrace{\phi(x_{11}) \circ \phi(x_{12}) \circ \phi(x_{13}) \circ \dots \circ \phi(x_{hw})}_{\text{tensor of order } hw} \hookrightarrow \mathbb{R}^{2^{hw}}$$

Two examples for ϕ are

- $\phi : \mathbb{R} \longrightarrow \mathbb{R}^2$

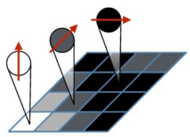
$$\alpha \longmapsto \begin{pmatrix} \alpha \\ 1 \end{pmatrix}$$

$$\hookrightarrow \quad (\alpha_1, \alpha_2, \alpha_3) \longmapsto \underbrace{\phi(\alpha_1) \circ \phi(\alpha_2) \circ \phi(\alpha_3)}_{} = \begin{pmatrix} \alpha_1 \\ 1 \end{pmatrix} \circ \begin{pmatrix} \alpha_2 \\ 1 \end{pmatrix} \circ \begin{pmatrix} \alpha_3 \\ 1 \end{pmatrix}$$

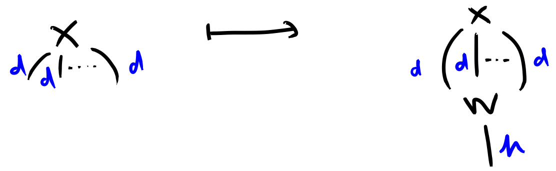
Entries of this tensor : $1, \alpha_1, \alpha_2, \alpha_3, \alpha_1\alpha_2, \alpha_1\alpha_3, \alpha_2\alpha_3$ and $\alpha_1\alpha_2\alpha_3$

- $\phi : \mathbb{R} \longrightarrow \mathbb{R}^2$

$$\alpha \longmapsto \begin{pmatrix} \cos\left(\frac{\pi}{2}\alpha\right) \\ \sin\left(\frac{\pi}{2}\alpha\right) \end{pmatrix}$$



$$f : \mathbb{R}^{d^m} \approx \mathbb{R}^{\overbrace{d \times d \times \cdots \times d}^{m}} \longrightarrow \mathbb{R}^p$$

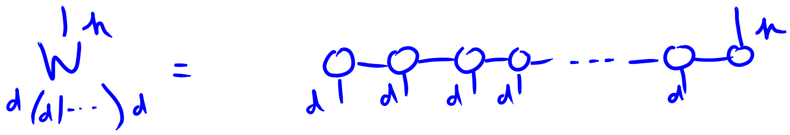


↳ The model parameter is $W \in \mathbb{R}^{d \times \cdots \times d \times p}$

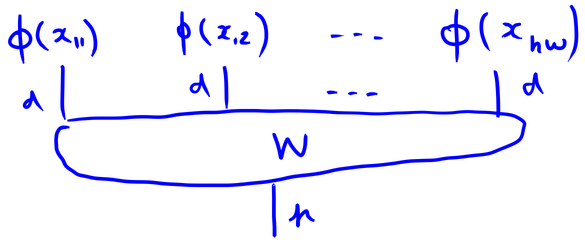
↳ $d^m p$ parameters !

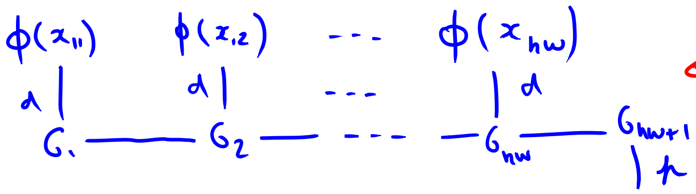
We cannot even store W in memory !

<u>Sol:</u> Parameterize W with a low rank TT decomposition



$$f(X) = f(\phi(x_{11}) \circ \phi(x_{12}) \circ \cdots \circ \phi(x_{hw}))$$





← We can compute this very efficiently

<u>Learning problem:</u> Given data $\{(X_1, y_1), \ldots, (X_m, y_m)\} \subseteq \mathbb{R}^{d^m} \times \mathbb{R}^p$

we want to minimize $\mathcal{L} = \sum_{i=1}^{m} \ell(f(X_i), y_i)$ for some loss function $\ell : \mathbb{R}^p \to \mathbb{R}$.

↳ parameters : $G_1, G_2, \ldots, G_{m+1}$ cores of a TT decomposition of W.

Training Algorithm (DMRG)

- Initialize $G_1, G_2, \dots, G_{m+1}$
- Repeat
  - for each consecutive pair of cores G_i, G_{i+1}
    - (merge two cores) $\quad {}_{R_{i-1}}\!-\underset{d\,/\,\backslash\,d}{B}-{}_{R_{i+1}} = {}_{R_{i-1}}\!-\underset{d|}{G_i}\overset{R_i}{-}\underset{d|}{G_{i+1}}-{}_{R_{i+1}}$
    - (Gradient descent step) $\quad B^{new} = B - \gamma \nabla_B \mathcal{L}$
    - (split in two cores) $\quad -\underset{/\,\backslash}{B^{new}}- \approx {}_{R_{i-1}}\!-\underset{d|}{G_i^{new}}\overset{R_i^{new}}{-}\underset{d|}{G_{i+1}^{new}}-{}_{R_{i+1}}$ * (truncated SVD)
    - $G_i, G_{i+1} \leftarrow G_i^{new}, G_{i+1}^{new}$
- until convergence

* R_i^{new} can be chosen adaptively from the singular values of $>B^{new}<$.

Follow up papers:

From probabilistic graphical models to generalized tensor networks for supervised learning

Ivan Glasser,[1,2] Nicola Pancotti,[1,2] and J. Ignacio Cirac[1,2]

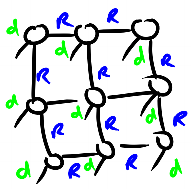
Tensor Networks for Probabilistic Sequence Modeling

Jacob Miller	Guillaume Rabusseau	John Terilla
Mila and DIRO	Mila and DIRO	CUNY and Tunnel
Université de Montréal	Université de Montréal	City University of New York
Montréal QC, Canada	Montréal QC, Canada	New York NY, USA
jmjacobmiller@gmail.com	grabus@iro.umontreal.ca	jterilla@gc.cuny.edu

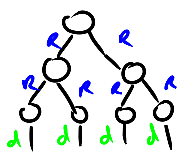
IV BEYOND TUCKER, TT, ...

TT :

PEPS (2dimTN) :



HIERARCHICAL TUCKER : (Tree TN)



TENSOR RING :